FlexNGIA
A Fully Flexible Novel Architecture for the Next-Generation Tactile Internet

Mohamed Faten Zhani

École de technologie supérieure (ÉTS Montreal)

University of Quebec

Canada

Beijing, China, 19 August 2019

Outline

- A Glance into the Future

- Limitations of Today's Internet

- FlexNGIA: Fully-Flexible Next-Generation Internet Architecture

- Use cases

- Conclusion

A Glance into the Future
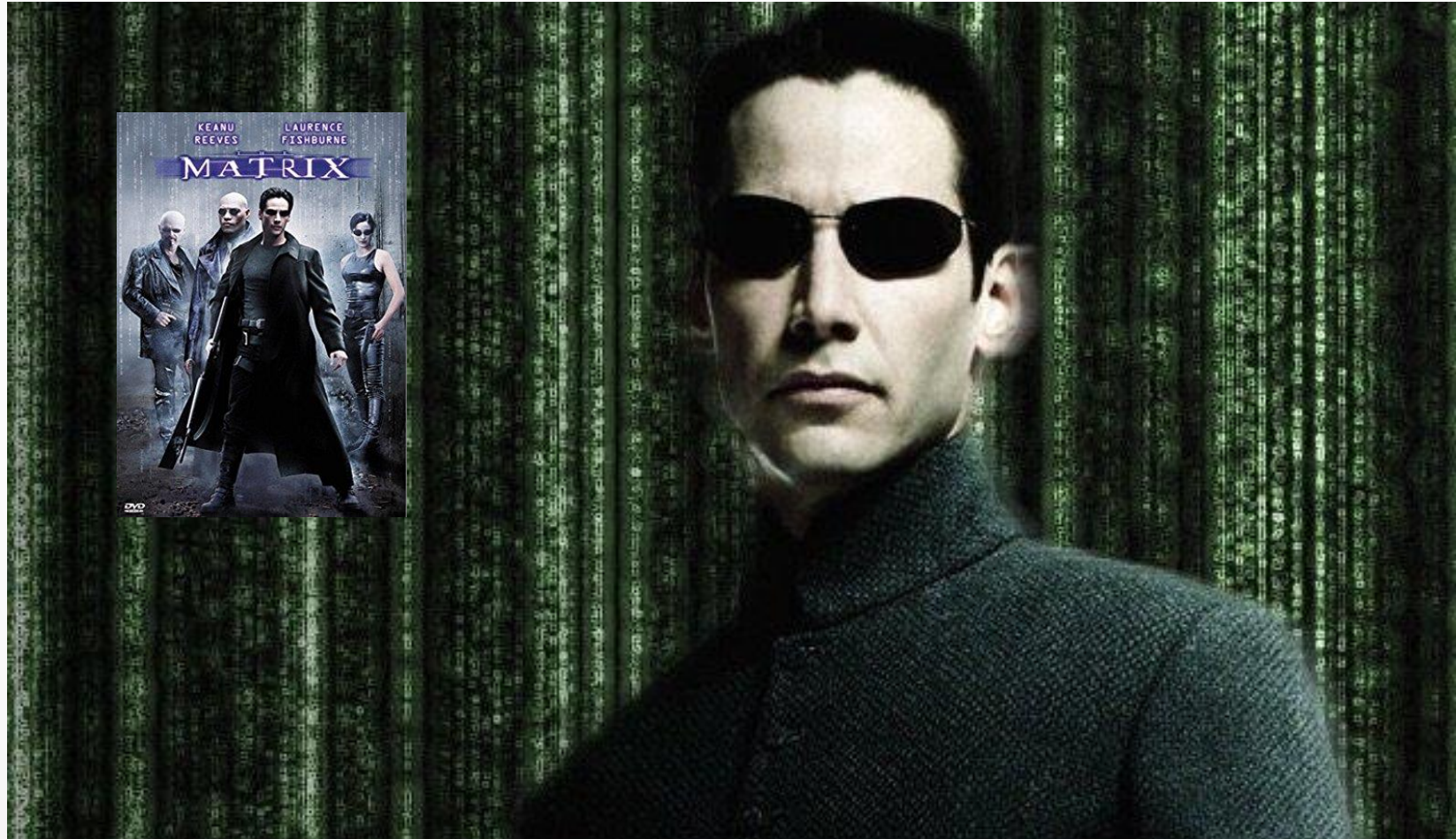
Future Applications

- Telepresence

- Virtual Reality

- Augmented Reality

- Holoportation

- Haptics

- …

Loading the Matrix…

Welcome to the Matrix

Future Applications Requirements & Characteristics

- ## Characteristics
  - Octopus-like applications: huge number of flows for each application
  - Changing requirements : requirements can change over time

- ## Requirements:
  - High processing power: real-time processing
  - High bandwidth (e.g., VR (16K, 240 fps) ➜ 31.85 Gbps)
  - Ultra-low Latency: 1ms to 20ms
  - Multi-flow synchronization
 - High availability

Outline

- A Glance into the Future

- Limitations of Today's Internet

  o Internet Infrastructure and Services

  o Network Stack Layers and Headers

- FlexNGIA: Fully-Flexible Next-Generation Internet Architecture

- Use cases

- Conclusion

Internet Infrastructure and Services

- A network of networks

- Offered service: "Best effort" data delivery.. no more

- No control over the infrastructure

  ➔ No control over the end-to-end path and quality of service

  ➔ No performance guarantees
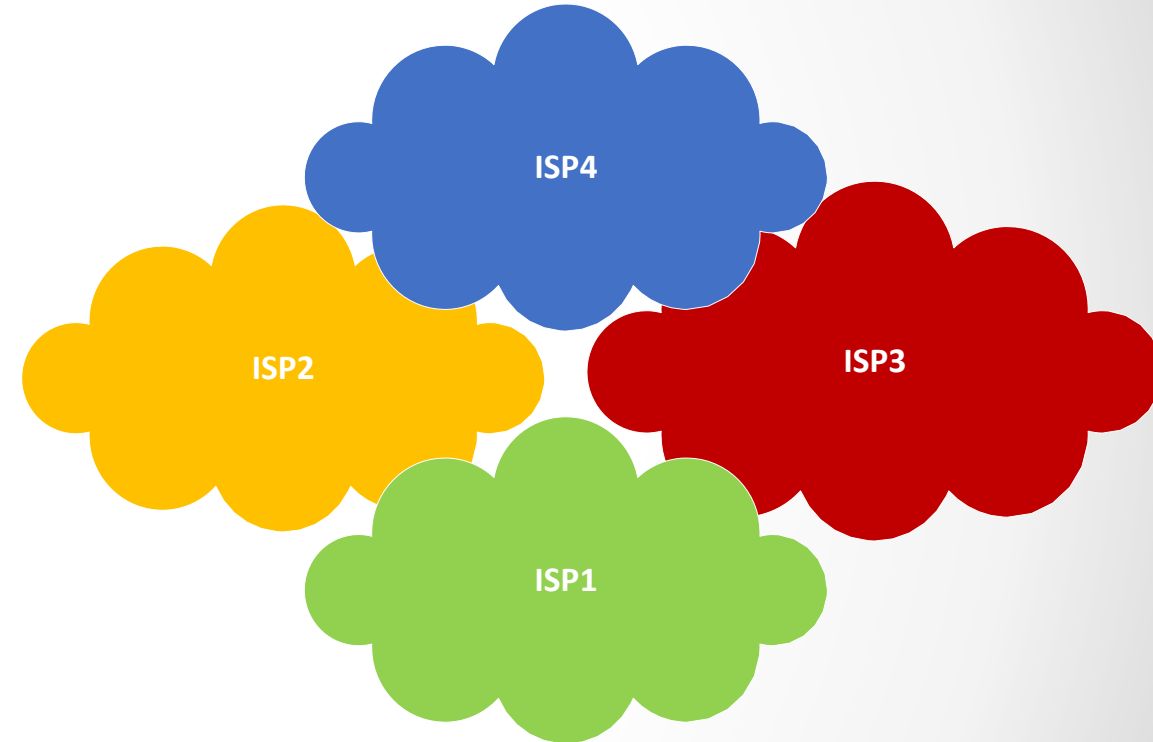
Transport Layer Protocols

Many modern protocols like SCTP and QUIC but let's focus first on TCP..

- One-size-fits-all service offering: TCP offers reliability, data retransmission, congestion and flow control

- Blind Congestion control

- The two end points limitation:
  - High retransmission delays (~ 3x e2e delay)
  - Transport and network layers are not aware which flows belong to the same application

Network Layer Protocols

- Not aware of the applications

  o The application composition (in terms of flows)

  o Performance requirements of each of these flows and how these requirement change over time

  ➔ Drop packets « blindly »

- No collaboration with the transport layer

  o Do not provide explicit feedback or support to transport layer (maybe ECN is interesting but it is not enough)

  o Do not help with other transport services (e.g., reliability)

Network Stack header

Problems with current headers:

- Do not provide additional informations about objects/sensors, flows belonging to the same application, applications' requirements, etc.

- Not flexible enough: It is not easy to incorporate meta-data and commands

Outline

- A Glance into the Future

- Limitations of Today's Internet

- FlexNGIA: Fully-Flexible Next-Generation Internet Architecture

  - Future Internet Infrastructure and services
  - Business Model
  - Management Framework
  - Network Protocol Stack/Functions
  - Stack Headers

- Use cases

- Conclusion

How a network will look like?

- Computing resources are everywhere: Available at the edge and at the core of the network

- Commodity servers but also dedicated hardware, FPGA, GPU, NPU, etc.

  ➔ In-Network computing

  ➔ Reduce steering delay

  ➔ Full Programmability: Any function could be provisioned anywhere (virtual machines/containers)

Cloud Data Center
Micro cloud
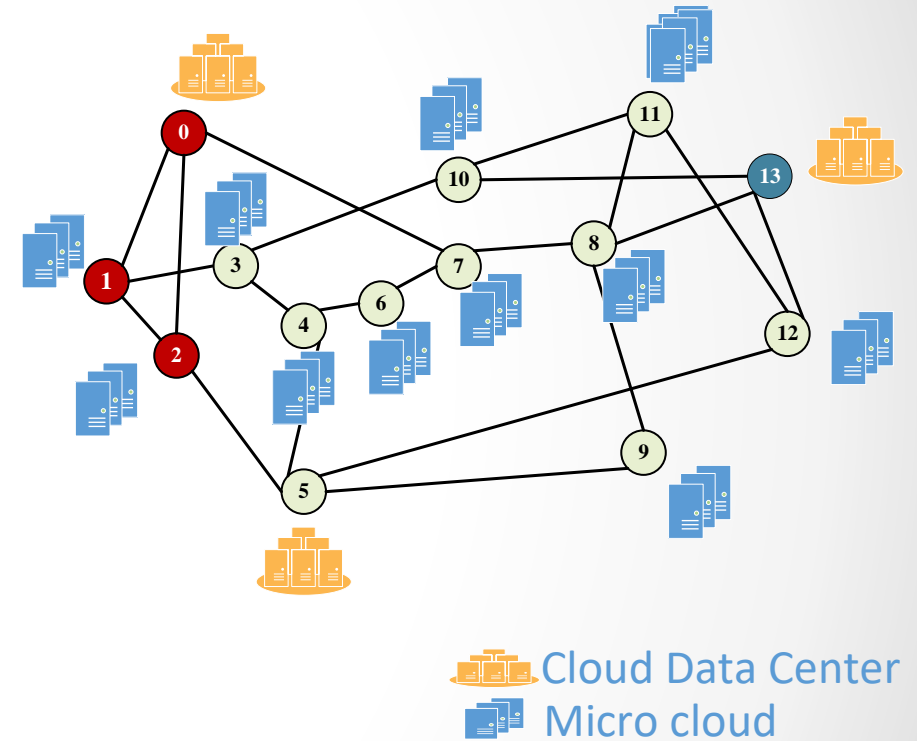
Future Internet Infrastructure and Services
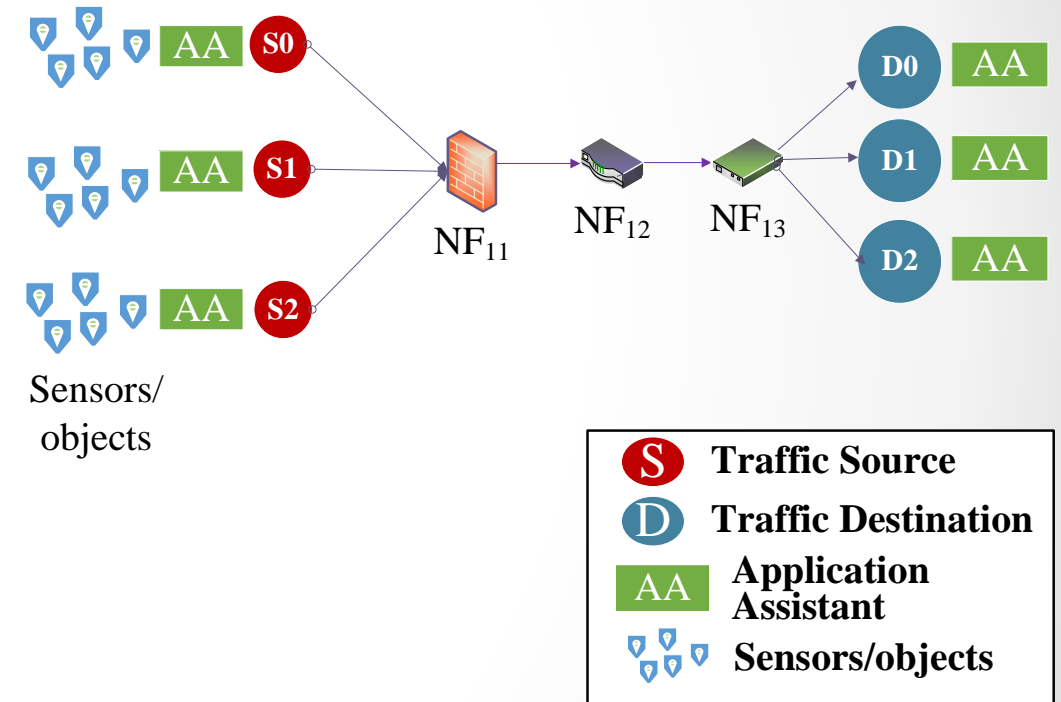
How does Future Internet look like?

- Still a network of networks..

- What is new?

    o More services: Service Function chains
    ➔ More advanced functions
    ➔ More than just delivery

    o Stringent performance guarantees

Future Internet Infrastructure and Services

## Service Function Chain (SFC)

- Multiple connected network functions

- Multiple sources and destinations

- Made out from Network Functions

- Defines, for each network function, the type, software, input/output packet format, expected processing delay, buffer size

- Defines performance requirements (e.g., throughput, packet loss, end-to-end delay, jitter)

Sensors/objects

S	**Traffic Source**
D	**Traffic Destination**
AA	**Application Assistant**
	Sensors/objects

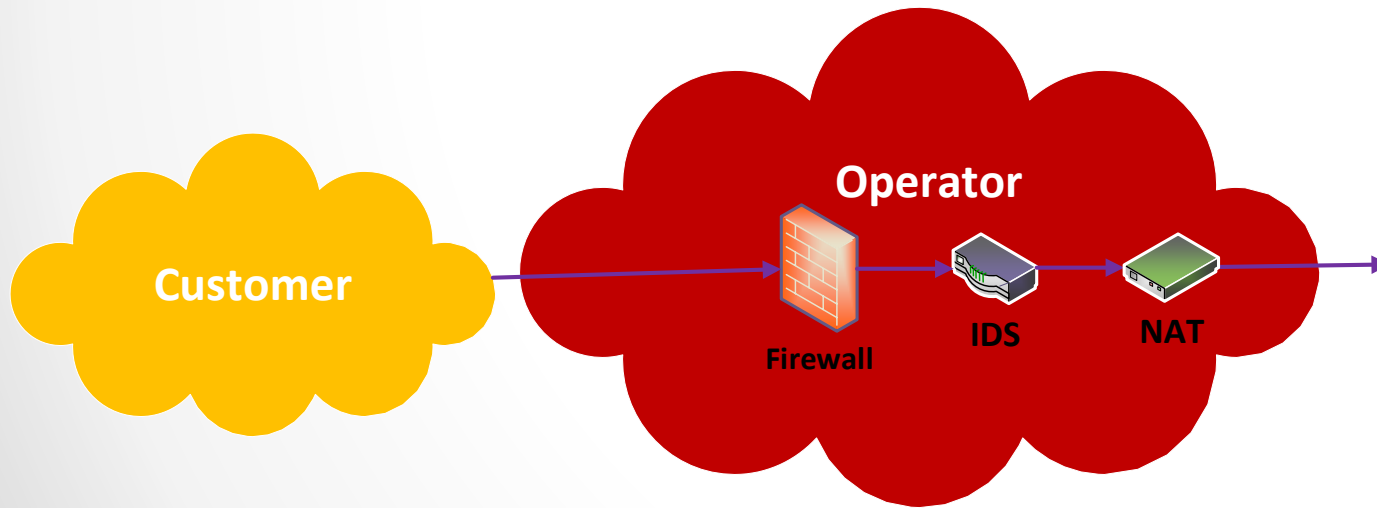# Business Model

Network Operators

- Own and manage the physical infrastructure (i.e., one network)

- Deploy platforms and software required to run network functions

- The service could be simply data delivery or a SFC

- Provision and manage SFCs
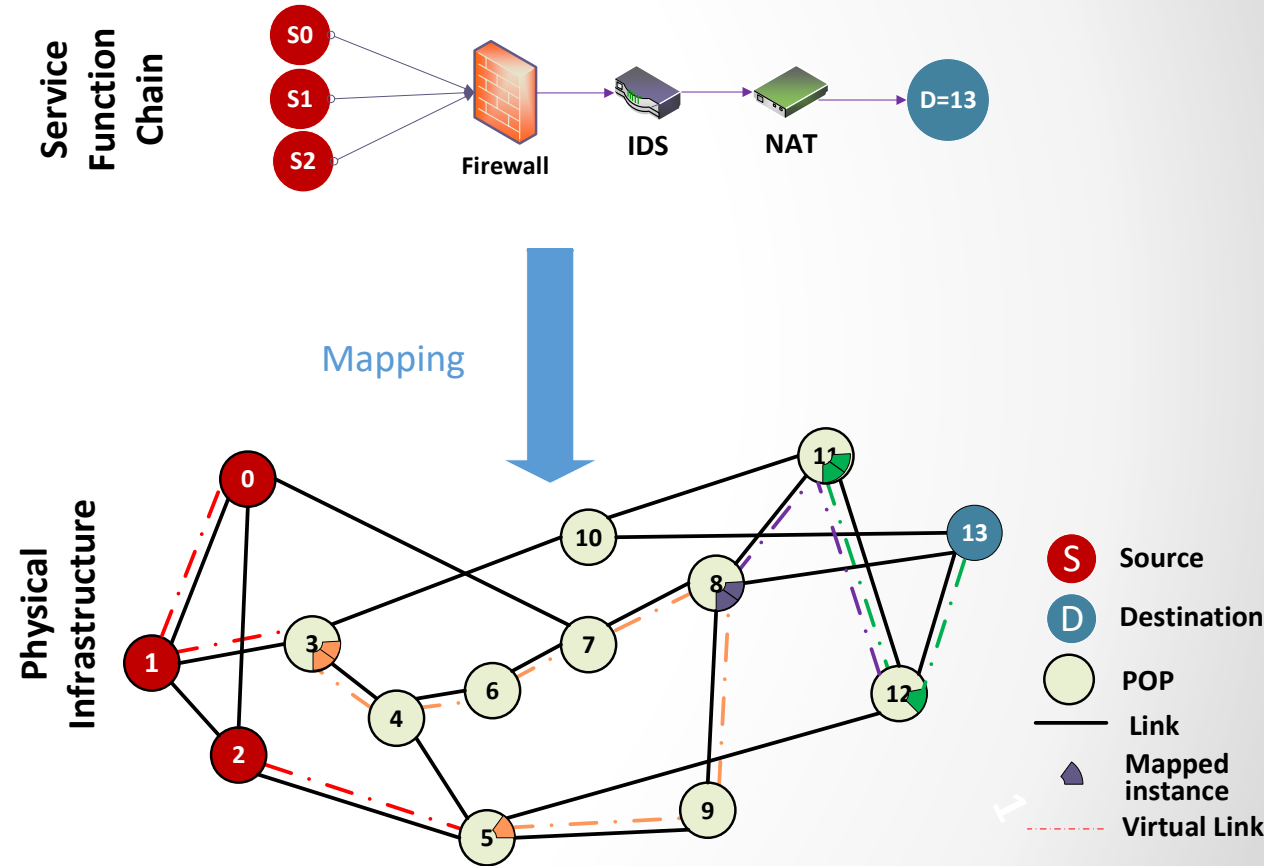
Business Model (cont)

Customers

- Could be other network operators, companies or Institutions

- Define the required SFC and Identify the chain sources/destinations

- Rely on the operator to provision and manage the SFC and satisfy SLA

Operator

Customer

**Firewall**   **IDS**   **NAT**

- SFC composition

- SLA requirements for the SFC
  - Bandwidth
  - End-to-end delay
  - Reliability, availability

- SLA requirements for each NFs
  - Processing power
  - Packet format(s)
  - Packet drop criteria...

Business Model (cont)

- Example of potential Network Operators:
  - ISPs (e.g., AT&T or Bell Canada) and web-scale companies (e.g., Google, Facebook, Amazon)
  - Example: Google Cloud Platform
    - World wide global Infrastructure
    - Software defined platform
    - Full control over the infrastructure

🔴 15 Data centers

🔵 100 Points of Presence (PoPs)

🟡 1000+ Edge nodes

Source: cloud.google.com
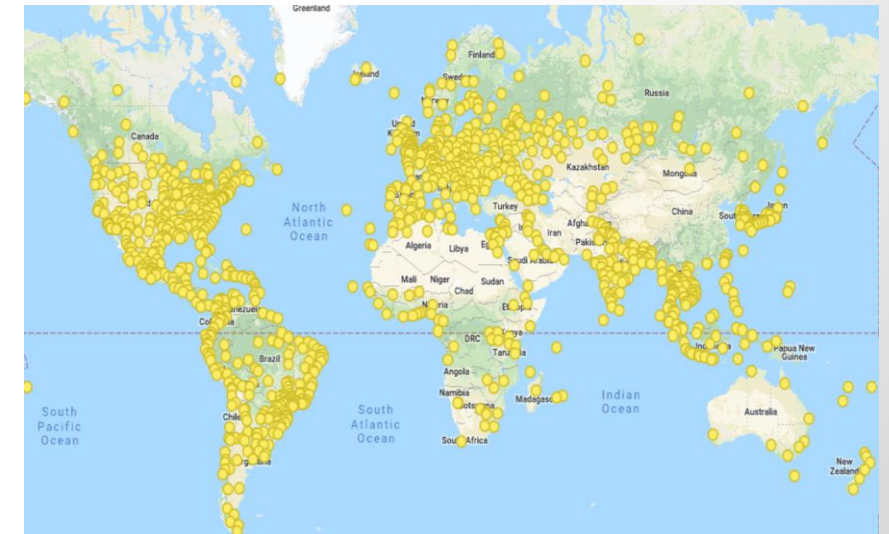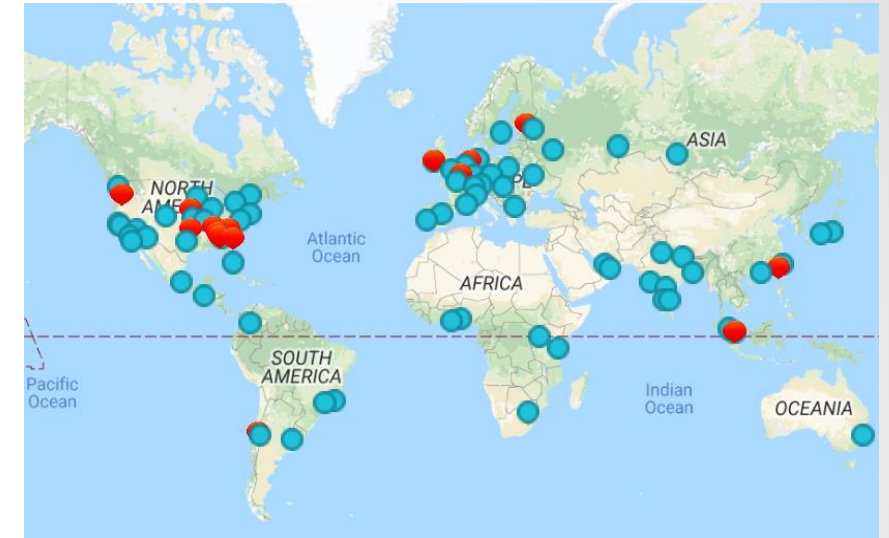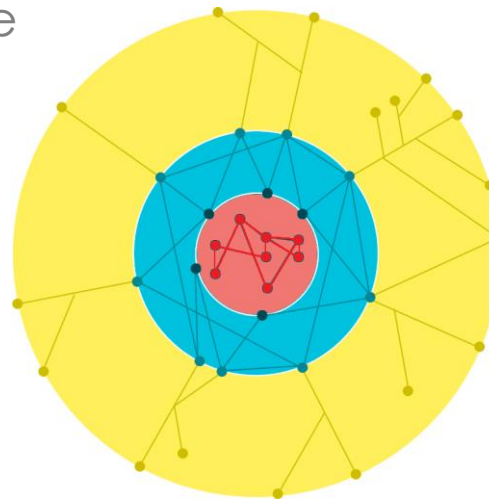
Resource Management Framework

Resource Allocation

- The Service Function Chain (SFC) is defined by the application designer

- 2-step resource allocation:
  - Translation: the SFC is translated into a virtual topology
  - Mapping: virtual topology are mappa
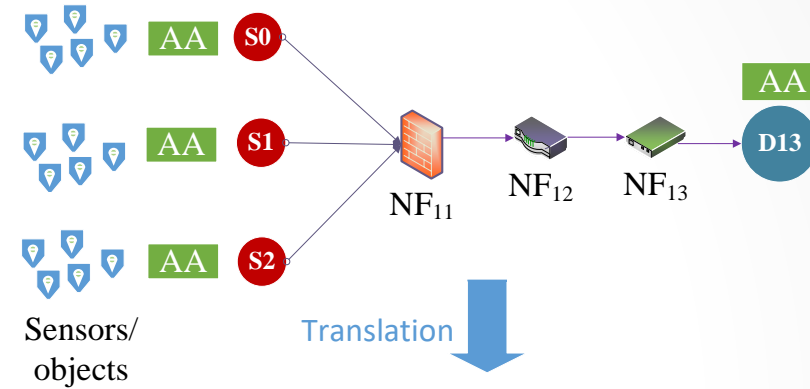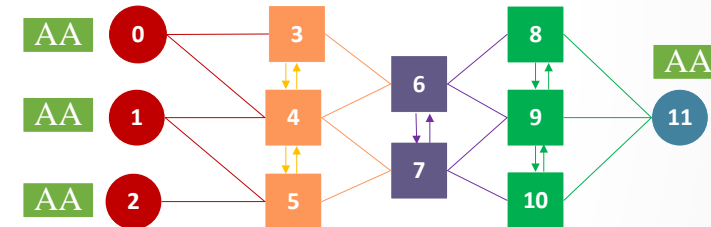
Resource Management Framework

Main components:

- Signaling module
- Application Control Module
- Ressource allocation Module
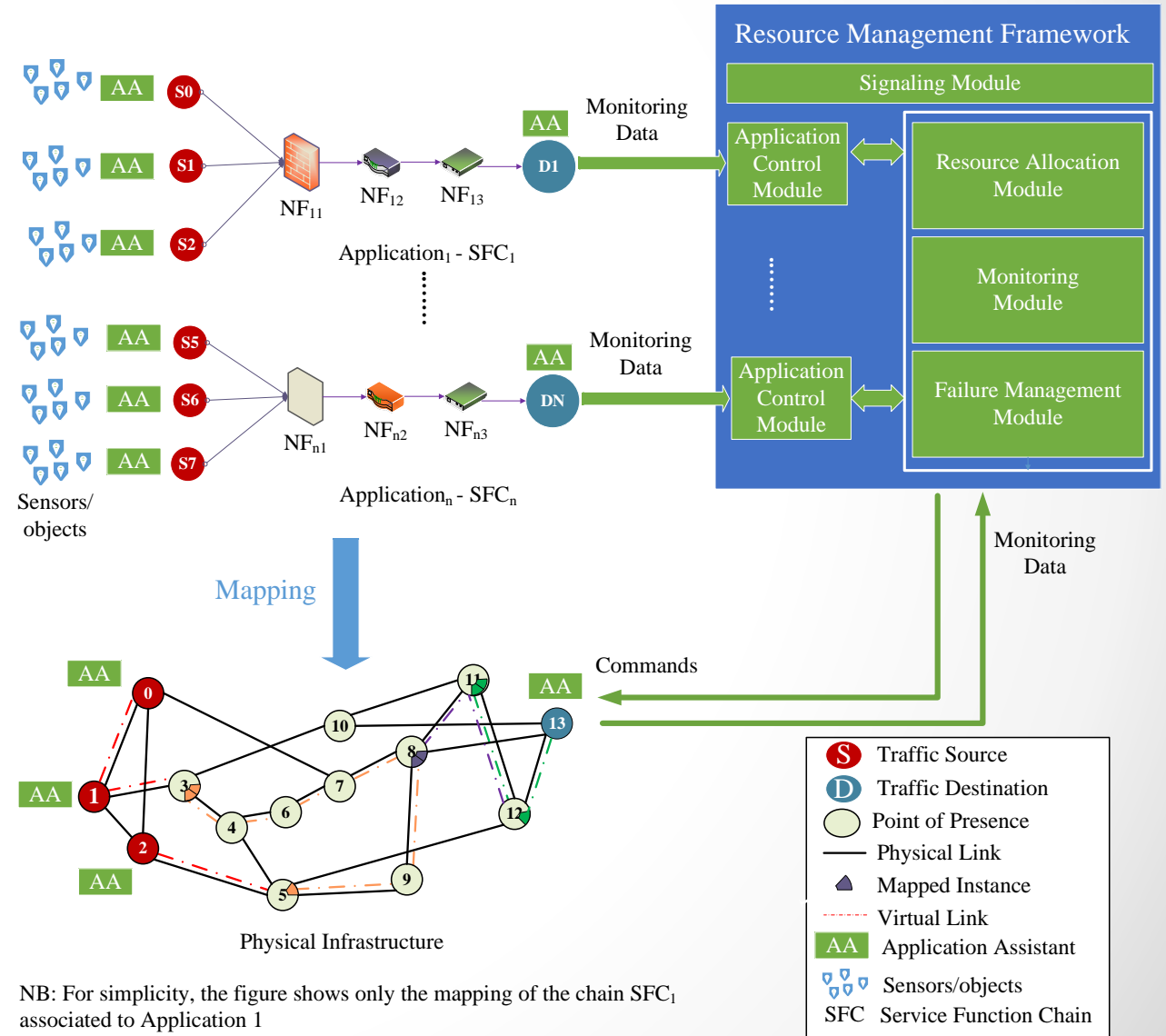
NB: For simplicity, the figure shows only the mapping of the chain SFC_1 associated to Application 1

Network Protocol Stack/Functions

- Basic Network Functions (e.g., packet forwarding)

- Advanced Network Functions:
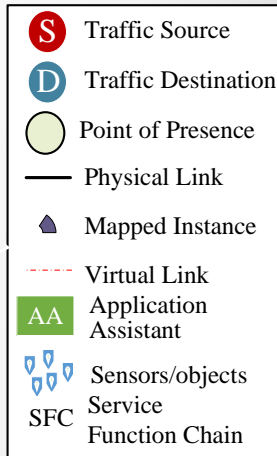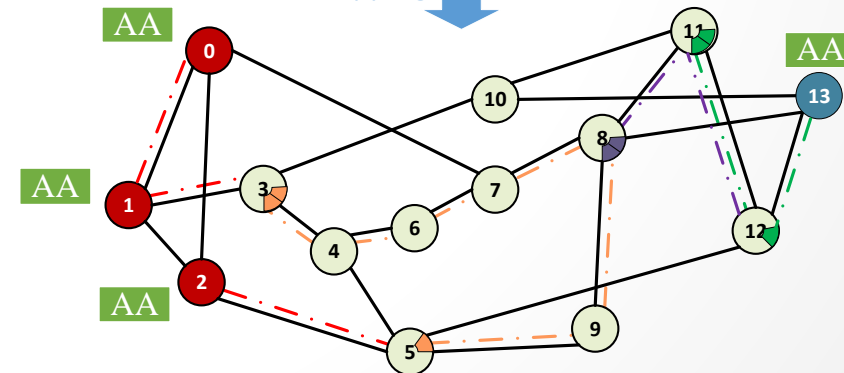 - Could operate at any layer
  - Only limited by our imagination

  - Examples: packet grouping, caching and retransmission, data processing (e.g., image/video cropping, compression, rendering, ML), application-aware flow multiplexing (e.g., incorporating/merging data)

→ Functions could break the end-to-end principle

→ SDN++: SDN should go beyond configuring forwarding rules and should provide the ability to dynamically configure these new functions
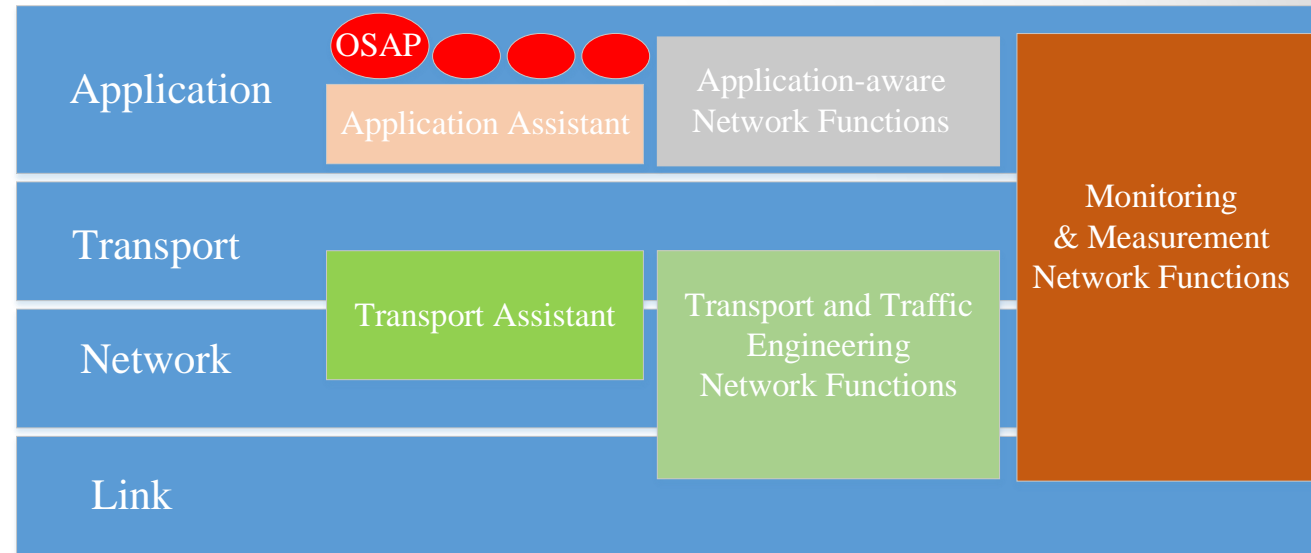
Application	OSAP ● ● ● Application Assistant	Application-aware Network Functions	Monitoring & Measurement Network Functions
Transport	Transport Assistant	Transport and Traffic Engineering Network Functions	
Network			
Link			

Application Assistant (AA)

- One AA at each end-point

- Interfaces with objects/sensors

- Measures the application performance and user QoE

- Identifies the applications' requirements at run-time

- Adds additional metadata To be used by subsequent Network Functions

→ Application-Aware Network Services

Sensors/objects

S — Traffic Source
D — Traffic Destination
AA — Application Assistant
Sensors/objects
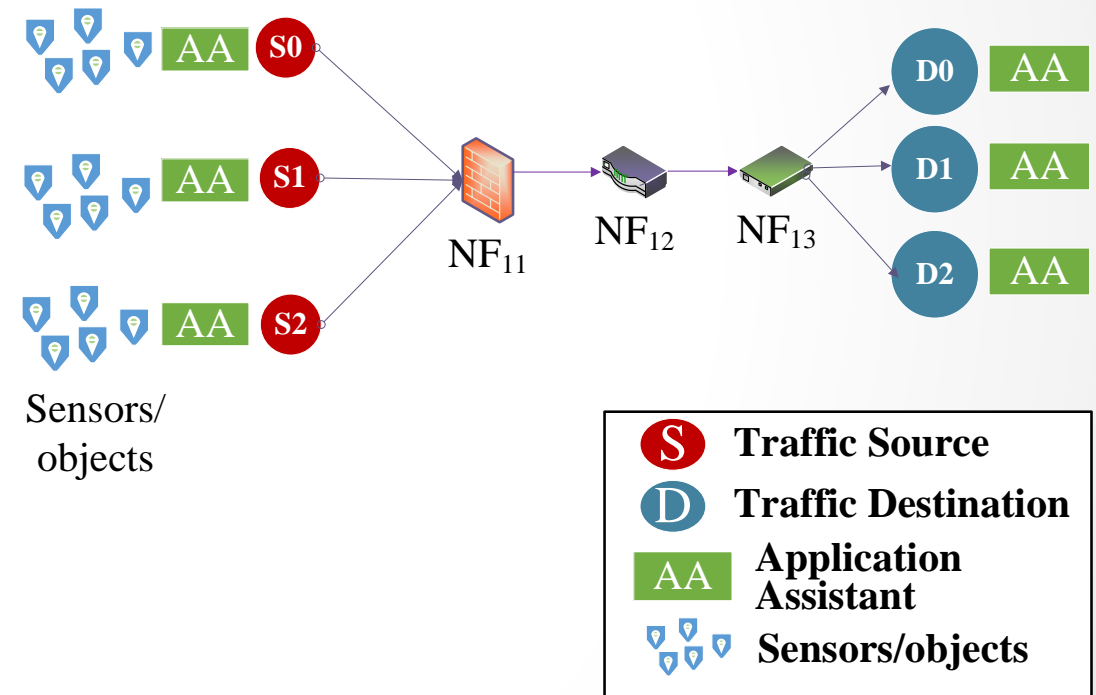
Network Protocol Stack/Functions Transport Assistant

Transport Assistant (TA)

- A cross-layer Network Function

- Combines services of the transport and network layers

- Manages all the flows of the same application

- Implements Transport/Network functions (e.g., congestion control, packet loss detection, packet cache and retransmission, routing)

- One or multiple TA could be provisioned in the same SFC

Application Layer

Transport Layer (TCP)
- E2E communication
- Blind congestion Control
- Inaccurate Packet Loss Detection
- Guaranteed Reliability
- E2E Packet Retransmission Process

Network Layer
- IP protocol (header and addressing)
- Routing Protocols/SDN
- ICMP for Control Information
- No Advanced Network Functions

Cross-Layer Transport
- Multi-point communication
- Network-assisted congestion control
- Network-assisted reliability and performance guarantees
- Accurate packet loss detection
- Variable performance and reliability Requirements over time
- Variable Header
- Meta-data and commands within packet headers
- Advanced Network Functions

Link Layer

Network Stack Headers

- Signaling packets
  - Instantiate an application
  - Convey application requirements

- Data packets: carry data
  - Layer 2 header: contains mainly the application id used for packet forwarding (similar to VLANs)
  - Upper layers:
    - Fully flexible header format (customizable meta-data and commands)
    - Defined depending on the application
    - Network functions should be aware of the expected format

Outline

- A Glance into the Future

- Limitations of Today's Internet

- FlexNGIA: Fully-Flexible Next-Generation Internet Architecture

- Use Cases scenarios

- Conclusion

Network-Assisted Data Transport

- Goal
  - Minimize retransmission delay
  - Improved congestion control

- Solution: service chain with a "transport Assistant" function

- Service of the Transport Assistant:
  - Caching and retransmissting packets
  - Detecting packet loss
  - Congestion control: adjusting rate, dropping packets, compression

Network-Assisted Transport

S0 → Transport Assistant → D=13

Mapping

Physical Infrastructure

	S	Source
	D	Destination
		POP
	—	Link
	▲	Mapped instance
	----	Virtual Link
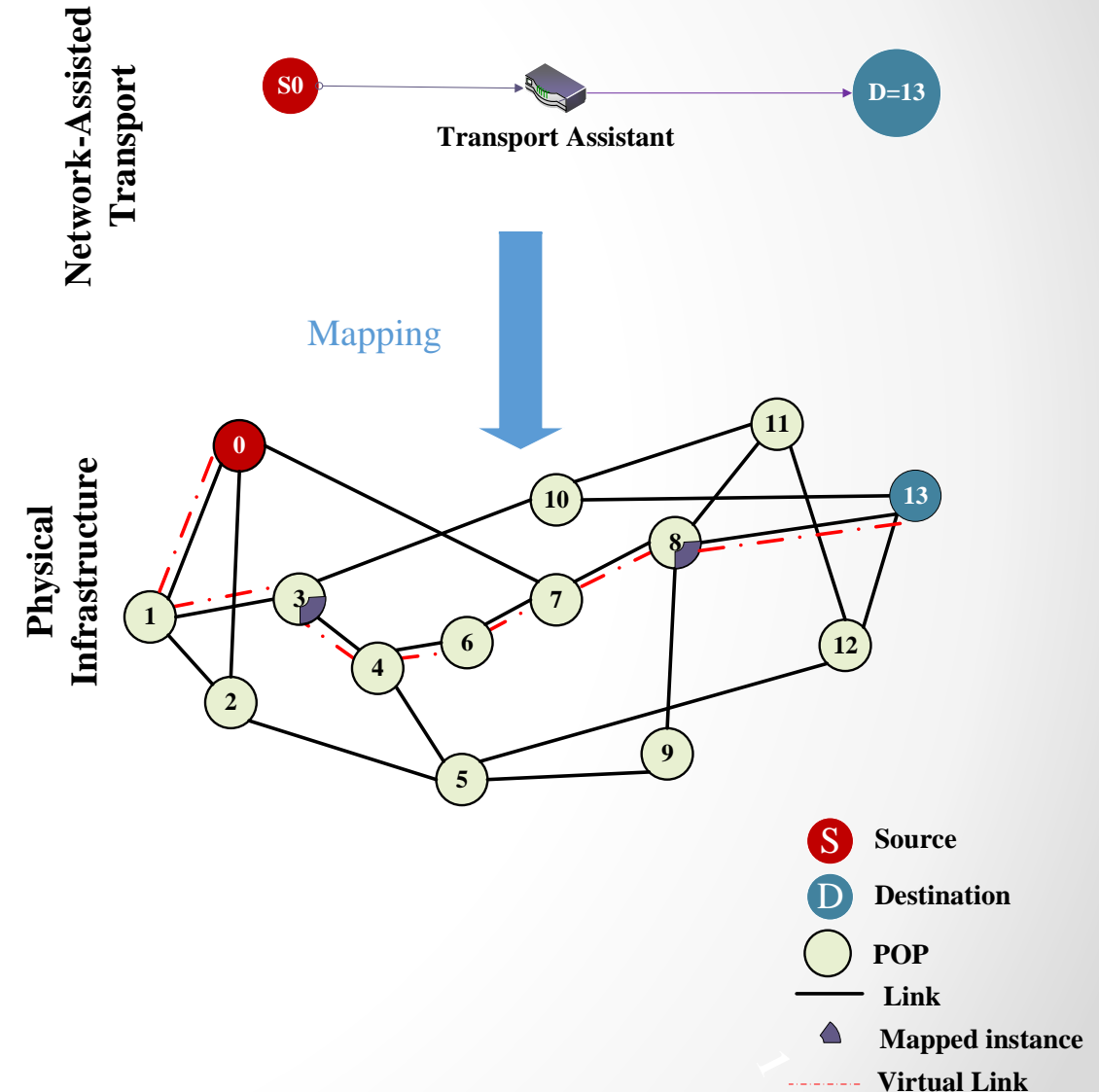
Mixed Virtual Reality and Holograms

- Users are exploring a virtual reality environment with several human holograms and objects

- Challenges
  - How many intermediate functions?
  - What kind of functions?
  - How the traffic should steered from the flow sources?
  - How many instances for each function?
  - Where to place them?

- Example of deployement
  - Encoder: encode and compress video
  - Transport manager: congestion control
  - Video cropper: crop 3D objects

Outline

- A Glance into the Future

- Limitations of Today's Internet

- FlexNGIA: Fully-Flexible Next-Generation Internet Architecture

- Use Cases

- Conclusion

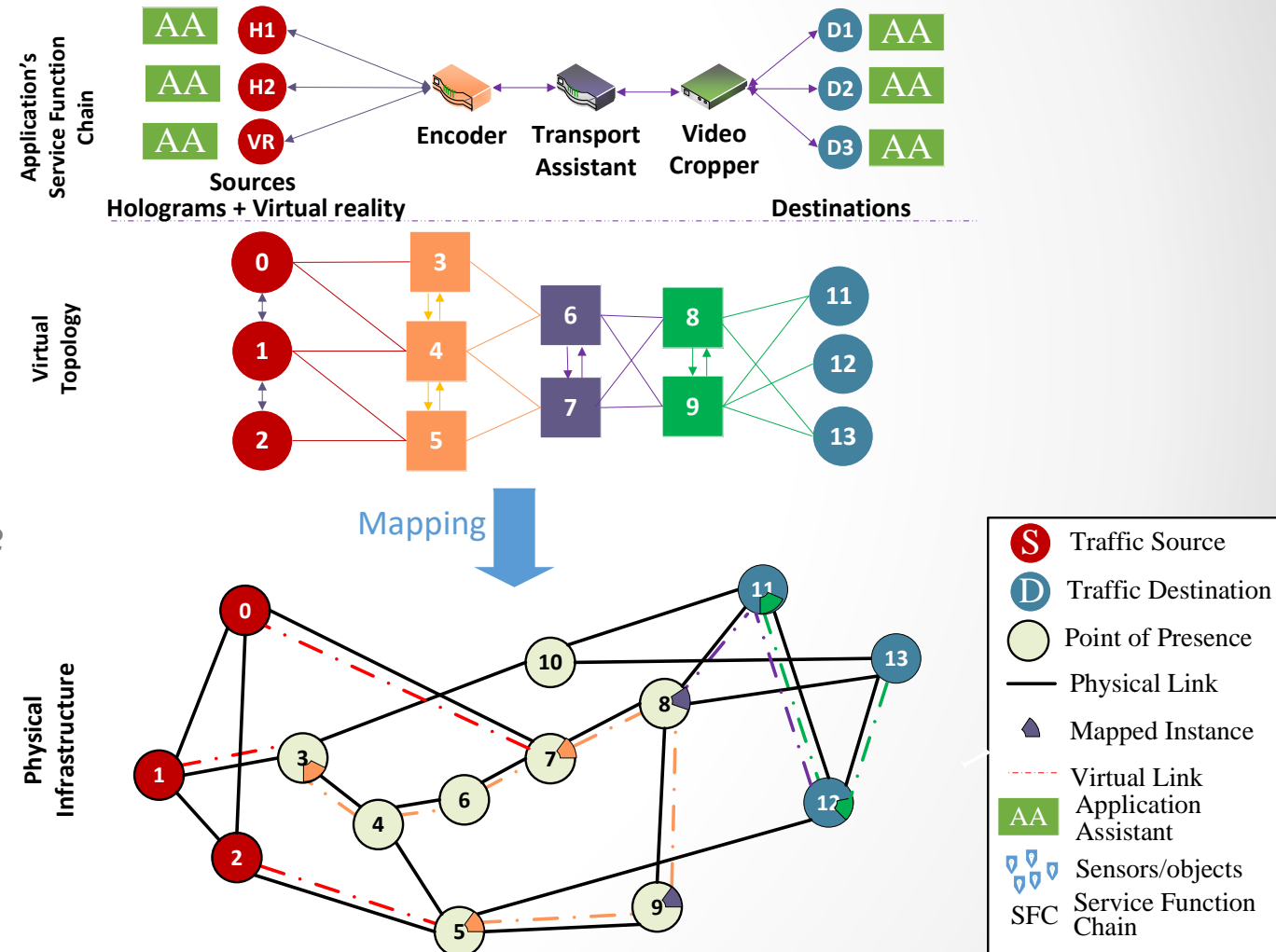# Conclusion

FlexNGIA

Computing resources	Business model	Cross-layer Design (Transport+Network)	Application-Aware Network Management	Flexible headers
- In-Network Computing: any function anywhere	- Multiple source destination Service Function Chains - Stringent performance requirements	- Breaking the end-to-end paradigm - In-network advanced transport functions - Better congestion control - Stringent performance and reliability guarantees	- Advanced functions tailored to applications - App-aware traffic engineering	- Tailored to the application

Looking for More Details?

- Mohamed Faten Zhani, https://profs.etsmtl.ca/mfzhani/

- FlexNGIA Project: M. F. Zhani, H. ElBakoury, "*FlexNGIA: A Flexible Internet Architecture for the Next-Generation Tactile Internet,*"ArXiV 1905.07137, May 17, 2019 https://arxiv.org/abs/1905.07137

École de technolgie supérieure (ÉTS Montreal)
University of Quebec, Canada

Thank You

Questions

Research Challenges

- Designing Service Function Chains tailored to applications

- High-performance softwarized functions

- Signaling

- Resource Allocation

- Fault-tolerance and Failure Management

- High-Precision and Fine-grained Monitoring and Measurements

- SDN++

- Distributed Cross-Layer Transport Protocol (sockets, caching, communication)

- Security and Privacy

Details are available in the paper (https://arxiv.org/abs/1905.07137)

Transport Layer Protocols (cont)

QUIC

- Transport over UDP
- Multi-streaming:
  - Every stream is a **reliable** bidirectional bytestream
  - Multiplexed streams between **two endpoints**
  - Stream prioritization
- Flow-control and congestion control very similar to TCP
- Endpoints use Explicit Congestion Notification (ECN)

SCTP

- Basically, a TCP++
- Multi-streaming
- Unordered delivery is possible
- Flow control and congestion control similar to TCP

Transport Layer Protocols (cont)

What are the limitations of SCTP and QUIC?

- E2E communication: multiple flows (streams) of the same application may connect more than two end-points

- A blind congestion control

- No support from the network: the network knows better about its state

  ➔ Can better locate and manage congestion

  ➔ Predict and detect more efficiently congestions/failures/problems…

  ➔ Can retransmit faster

  ➔ Can provide better garantees in terms of delay and packet loss

Network Protocol Stack/Functions Transport Assistant (cont)

(a) TCP and UDP   (b) SCTP and QUIC   (c) FlexNGIA

- Physical end-host
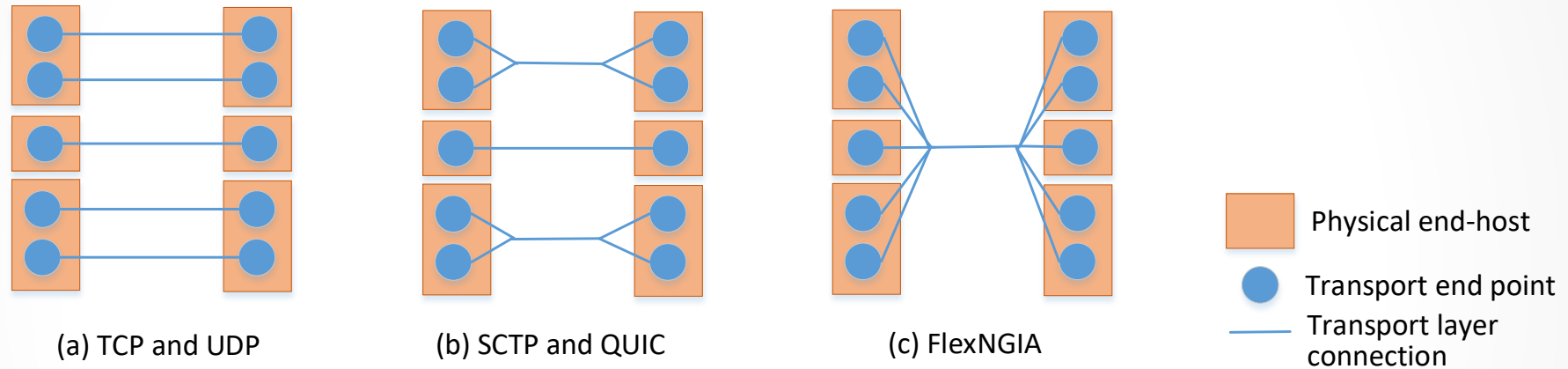- Transport end point
- Transport layer connection

Illustration of how one single network application might be seen at the transport Layer

- Transport Assistants manage all these flows while taking into account that they all belong to the same application

- TAs monitor these flows, divide the total bandwidth allocated for the application among them.